Project Setup and Backup

02/06/2025 14:08

Perfect — that structure breakdown from **`` `diagnose_structure_sources.py` ``** is exactly what we need. Here's the plan:

✅ Curated Test Bench: 6 PDFs, Each with a Different Structure Mode

Structure Type	Suggested PDF	Purpose
`` `headings` ``	**5e_2024_House_Rules_draft_v_0.2.pdf**	Validate heading detection + fallback
`` `outline` ``	**Classic_Guide_to_Greyhawk.pdf**	Outline structure matching
`` `visual` ``	**Greyhawk_Expanded_-_A_Players_Guide_to_Oerik.pdf**	Test visual TOC fallback
`` `mixed` ``	**VER1-03_Gift_of_Beauty_(3E).pdf**	Confirm behavior with outline + visual TOC
`` `no structure` ``	*(inject dummy or broken file)*	Simulate fallback to raw `` `.txt` ``
`` `already_cleaned` ``	**Greyhawk_Rebooted_cleaned.md**	Idempotency test for post-processors

---

🧪 Test Bench Directory Plan

```
data/
└── testbench_input/
    ├── 5e_2024_House_Rules_draft_v_0.2.pdf
    ├── Classic_Guide_to_Greyhawk.pdf
    ├── Greyhawk_Expanded_-_A_Players_Guide_to_Oerik.pdf
    ├── VER1-03_Gift_of_Beauty_(3E).pdf
    ├── Greyhawk_Rebooted_cleaned.md
    └── broken.txt
└── testbench_output/
```

📄 TEST_PLAN.md (to be created)

Each test will:

- Specify which script should run
- What input/output file to expect
- What *kind* of transformation should be visible (e.g., spacing cleanup, heading tagging, fallback behavior)

🛠️ Next Step:

Would you like me to:

1. Create the folder structure & `` `TEST_PLAN.md` `` scaffold?
2. Generate a helper script `` `run_testbench.py` `` to automate testing these scripts?
3. Help you copy these test files from your `` `converted_md` ``/`` `input_pdfs` `` dirs?

Ready when you are.